



INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY
CAMPUS QUERÉTARO

Laboratorio de desarrollo de aplicaciones web

Estándar de Codificación y Branching model

Ulises Almaguer Guzmán A01209070
Francisco Javier Castillo Hernández A01208970

14 de Septiembre de 2019

Periodo Ago-Dic 2019

Erick de Jesús Sánchez M.

Definición del Estándar de codificación

La forma de escribir código es propia de cada programador y completamente diferente a la forma de cualquier otro. De la forma usada depende la facilidad para entender el código y retomar ciertas partes realizadas por otros integrantes, así como la depuración de las mismas.

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software.

La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Aunque la legibilidad y la mantenibilidad son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación.

Estándar de Codificación 8970_9070

1) Nombre de variables:

- Los nombres que se usen deben de ser significativos
- Los nombres deben de estar en minúsculas, excepto la primera letra de cada palabra a partir de la segunda palabra. Ejemplo:
nmroPersonas
- El uso de variables temporales pueden tener el nombre que se desee siempre y cuando solo se use en las siguientes líneas y no en el resto del programa.

2) Nombre de registros

- Cuando se necesite obtener información de la base de datos por tupla se deberá de nombrar conforme \$row y el nombre de la tabla para mayor entendimiento

3) Nombre de programa

- Cada nombre tiene que ser significativo a lo que hace, la extensión de php se tiene que usar en torno a la acciones que hace o en caso de cumplir con todo el caso de uso llamarlo de esa manera

4) Constantes y variables globales

- Evitar el uso de constantes numéricas sin mucho significado y en caso de generar una será completamente con mayúsculas.
- Evitar el uso de variables globales

5) Corchetes e indentación

- El uso de corchetes debe de ser congruente por ejemplo en el caso de funciones,if,switch y diferentes estructuras el corchete inicio deberá de coincidir en espacio con el de fin.
- La indentación se deberá hacer con tabs y no espacios en blanco

6) Funciones

- Para las funciones es necesario llamarla en alusión a lo que hace y evitar nombres cortos que puedan malinterpretarse.
- En caso de ser funciones muy utilizadas o complejas escribir comentarios sobre lo que hace o a donde se pretenden llegar

7) Claridad de los programas

- Es necesario que se haga una descripción breve en archivos que se consideren de alta complejidad así como las funciones comentar en caso de que sea necesario

8) HTML y CSS

- Los tags tienen que estar escritos en minúsculas.
- Cuando hay que escribir variables de PHP dentro de un código HTML se debe tener el HTML dentro de PHP y no al revés
- Comprimir siempre el css

Selección de Branching model:

El modelo de control de versiones es el modelo de ramas de Driessen, el cual se describe a detalle a continuación, desde sus características más importantes, hasta sus desventajas y ventajas:

Git Flow:

Es el flujo de trabajo más conocido. Fue creado por Vincent Driessen en 2010 y se basa en dos ramas principales con una vida infinita:

- master: esta rama contiene código de producción. Todo el código de desarrollo se fusiona en master en algún momento.
- develop: esta rama contiene código de preproducción. Cuando las características están terminadas, se fusionan en desarrollo.

Durante el ciclo de desarrollo, se utilizan una variedad de ramas de soporte:

- features- *: las ramas de características se utilizan para desarrollar nuevas características para los próximos lanzamientos.
- release- *: las ramas de lanzamiento apoyan la preparación de un nuevo lanzamiento de producción. Permiten corregir muchos errores menores y preparar metadatos para un lanzamiento. Puede ramificarse del desarrollo y debe fusionarse en master y develop.

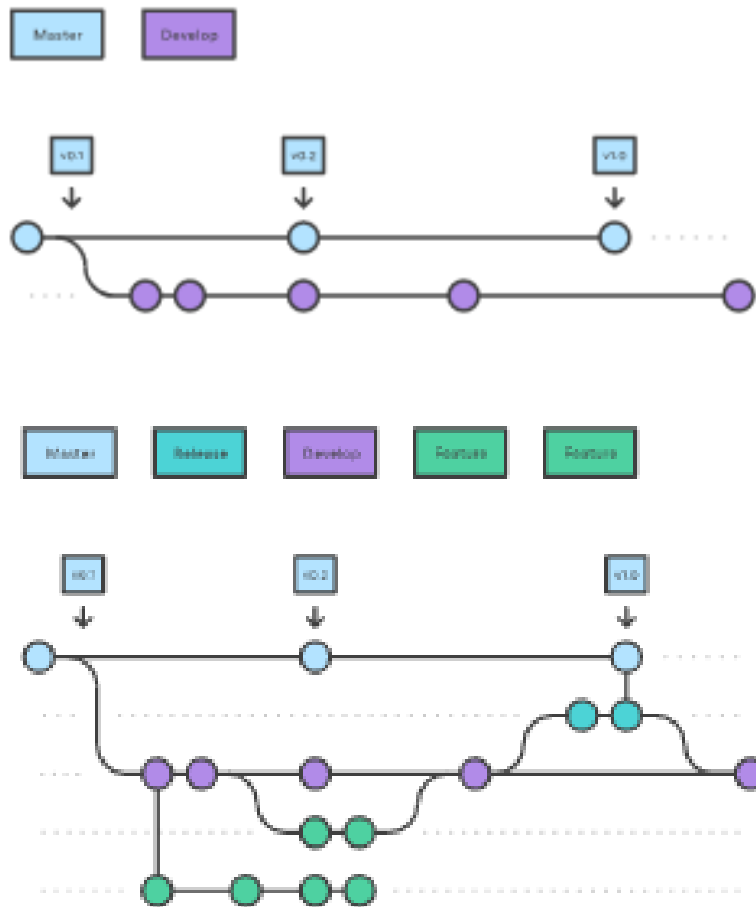
Ventajas

- Asegura un estado limpio de sucursales en cualquier momento del ciclo de vida de un proyecto.
- El nombre de las ramas sigue un patrón sistemático que hace que sea más fácil de comprender.
- Tiene extensiones y soporte en la mayoría de las herramientas git usadas.
- Es ideal cuando necesita una versión múltiple en producción..

Desventajas

- La historia de Git se vuelve ilegible.

- La división master / develop se considera redundante y hace que la entrega continua y la integración continua sean más difíciles.
- No se recomienda cuando necesita mantener una única versión en producción.



El flujo general de Gitflow es:

- Se crea una rama de develop desde master.
- Se crea una rama de lanzamiento a partir del develop.
- Las ramas de características se crean a partir del develop.
- Cuando se completa un feature, se fusiona en la rama de develop.
- Cuando se completa la rama de lanzamiento, se fusiona en develop y master.
- Si se detecta un problema en el master, se crea una rama de revisión desde master.
- Una vez que se completa la revisión, se fusiona en develop and master.